Deformable Convolution Networks

Arka Sadhu

IIT Bombay

October 18, 2017

Outline

Outline I

1. Deformable Convolutional Networks : Introduction

2. Deformable Convolutions
   - Deformable RoI and Deformable RoI pooling

Limitations of Convolutional Networks

- CNNs cannot model large unknown transformations because of fixed geometric structures of CNN modules.
- Convolution samples features at fixed locations.
- Region of Interest (RoI) use fixed spatial bins.
- Example : Receptive fields of a convolution layer is the same at all places. This is not desirable at higher layers which encode semantic features rather than spatial features.
- Instead of bounding boxes we would rather want exact boundaries.
- Hence we move on to Deformable Convolutional Networks.

Two New Modules

- Deformable Convolutions : basic idea is to add 2d offset to enable a deformed sampling grid. These offset are also learnt simultaneously along with the convolutional layers.
- Deformable RoI : similar idea. Adds offset to each bin position in the regular bin partitioning.
- Combined to get Deformable Convolutional Networks.
- Authors claim that this can directly replace existing CNN architecture.

Outline I

1. Deformable Convolutional Networks : Introduction

2. Deformable Convolutions
   - Deformable RoI and Deformable RoI pooling

Simple Convolution to Deformable Convolutions

- Let R denote the set of points which are to be considered for the convolution. In usual convolution of size 3 this R will have $(-1,-1)$ to $(1,1)$.
- Let input feature map be denoted by x and output feature map denoted by y, and w be in the weights of the convolution filter. For a particular point p_0,

$$y(p_0) = \sum_{p_n \in R} w(p_n)x(p_0 + p_n)$$

.

- For the case of deformable convolutions the new equation will be

$$y(p_0) = \sum_{p_n \in R} w(p_n)x(p_0 + p_n + \Delta p_n)$$

.

Simple Convolution to Deformable Convolutions (Contd.)

- Note: Δp_n can be fractional. To get the value of $x(p_0 + p_n + \Delta p_n)$ bilinear interpolation is used.
- Let G(., .) be the bilinear interpolation kernet. Then for any point p (could be fractional as well)

$$x(p) = \sum_q G(p, q) x(q)$$

- Authors claim that this is easy to compute since G will be non-zero at very small number of qs.

What is RoI and RoI pooling

- RoI is region of interest. The best example would be a bounding box for an object in an image.
- We would like to work even when this bounding box is not be constrained to rectangular.
- RoI pooling divides the RoI into k by k bins and outputs a feature map y of size k-by-k. This could be max or average pooling or any other kind of pooling. For say (i, j)-th bin with n_{ij} pixels we can have:

$$y(i,j) = \sum_{p \in bin(i,j)} x(p_0 + p)/n_{ij}$$

RoI pooling to Deformable RoI pooling

- For the deformable RoI pooling case we will instead have:

$$y(i,j) = \sum_{p \in bin(i,j)} x(p_0 + p + \Delta p_{ij)})/n_{ij}$$

- Again Δp_{ij} could be fractional and we would use bilinear interpolation.
- The paper introduces the idea of normalized offsets $\hat{\Delta p_{ij}}$ and actual offset is calculated using $\Delta p_{ij} = \gamma * \hat{\Delta p_{ij}} \cdot (w,h)$. This is intuitively required to account for the different k used in the RoI pooling. Emperically γ is set to 0.1